Reconceptualisation of Architects' Intentionality in Computational Form Generation: A Tripartite Model

Duygu Tüntaş

At the turn of the century, with the developments in computer science and increased capacity in information processing provided by the computational paradigm, studies on computational design display great interest in complexity management. The introduction and extensive use of computation and its associative thinking in the design process led to a great expansion of the dominant mode of computation – especially in form studies – that largely relies on data-driven forms as outcomes of pure calculations and rationalistic determinism. While the aim is to cope with the intricacy of data, as Zeynep Mennan informs us, the 'improved means and methods used in complexity management do not reduce but rather increase' the complexity of design problems.[1] In order to respond to the rapidly changing status of these technology oriented tools and mindsets, designers made an epistemic choice in favour of rationalisation with avoidance of subjectivity and its related modes of design thinking.[2] As an alternative to the increasing interest in this techno-rational tendency, this study proposes to understand and assess design intentionality by unfolding and thereby reflecting on designers' internalised processes.

Design intentionality

An investigation of design intentionality and its possible relationship with form computation is a great challenge. This study posits that such a link can be found in the interspace between the externalisation processes of design thought and their translation into a computational medium; it does so by examining the role of computation in the externalization of cognitive operations or ways of thinking that would be specific to the act of design, i.e. design behaviour. In an interview with Daniel Rosenberg, Humberto Maturana characterises design behaviour by conceptualising design operations and processes as *doing,* which is philosophically different from *being,* and describes design as an intentional act.[3] He asserts that

> the intentional act of design consists of manipulating the world that you live [in] … something will happen – in the flow of the changing cosmos that you are bringing forth with your living – so that you will be able to make a particular desired distinction and say, "this is what I wanted to do".[4]

He continues, 'as an intentional act, however, design specifies certain conditions of operation which will be the grounding conditions for something to happen, if those initial conditions are satisfied'.[5] Maturana argues that 'things are structure determined entities, so the task is to understand what is the organization, what is the structure, and what is the domain of variability.'[6] This means that design tools have structural dynamics inside, and the design operation is bound to the logic of that thing, i.e. its organisational capacity and coherence.[7] Like any other design tool, he considers computation as a tool for designers, very similar to what a brush is for an artist by emphasising its structural dynamics defining its operational capacity and the domain of variability.[8] From this viewpoint, there will be

conditions where the design tool may not respond to the designer's thought processes because of the incompatibility of the computational method's inner structural dynamics and underlying formal system with the design intentionality.

In *The Electronic Design Studio*, George Stiny points out the challenge created by the epistemological gap between the nature of the world of design and the 'structured' nature of computational world.[9] He states that 'designers do many things that computers don't. Some of these are bad habits that the stringencies of computation will correct. But others are basic to design, and cannot be ignored if computation is to serve creation and invention.'[10] He emphasises the importance of ambiguity in design to feed 'imagination and creativity' and to incorporate 'multilayered expression and response' into computational procedures.[11]

The dominant approach to form computing

In his foreword to Kostas Terzidis's *Expressive Form*, William Mitchell approaches the problem of dominant computational approach from a pragmatic-formal level.[12] Mitchell associates the formal tendencies with an '*economy of shapes*' – suggesting the availability and ease in the creation of some forms with certain methods – while the expansion and restructuring of these tendencies has been sustained with the advancements in computer technology.[13] Terzidis defines the same trend in the field of computational design, but this time from an epistemological perspective.[14] He notes that

> what makes computation so problematic for design theorists is that it has maintained an ethos of rationalistic determinism – the theory that the exercise of reason provides the only valid basis for action or belief and that reason is the prime source of knowledge – in its field. Because of its clarity and efficiency, rationalistic determinism has traditionally been a dominant mode of thought in the world of computation. The problem with this approach is that it assumes that all computational activities abide by the same principles. In contrast, intuition, as defined in the arts and design, is based on quite different, if not opposing, principles. … This mode of thought comes in contrast to the dominant computational model where methodical, predictable, and dividable processes exist.[15]

Terzidis reveals that the world of computation, in which a more rational, confined, organised, and methodical model exists, is resistant to such characterisations belonging to the human world, where 'intuition has been an underlying assumption for many design activities.'[16] Elaborating on this division, he claims that the mathematical processes can easily be translated into quantitative methods, thereby, can be controlled through computation, whereas 'manipulations, evaluations, and combinations of these processes are qualitative processes and as such can be handled by the architect.'[17] He notes that at the point where we shift our design modes from manual to computerised, it is necessary to 'integrate the two seemingly contrasting worlds, that of intuition and that of computation'.[18] The outcome of such reconciliation may provide an alternative to the dissolution of subjectivity.

As Terzidis points out, computational methods are argued to be rational because of their 'mechanistic nature,' and similarly, they are claimed as incapable of 'artistic sensibility and intuitive playfulness in their practice'.[19] In a similar way, Axel Kilian considers computation to be in many cases 'an obstacle … in translating design intent', since 'it lacks the fluidity of human thoughts'.[20] And he argues against this dominant view by stating that

> design should not be solely about the execution of established processes but about querying the understanding of the factors involved. This is a much more complex task and it goes far beyond the traditional geometric and numerical representation of current computational practices but it happens in designers' minds regardless of the involvement of computation.[21]

By extension, Kilian proposes to see the critique of the dominant approach to computational design not as 'a glorification of human designers' but as 'a reminder of the respective strengths and weaknesses of the different approaches', and not to perceive them as competing processes but as 'a potential collaboration between design in the mind and its externalised computational processes'.[22]

Based on the analytical and generative capacity of computational thinking, Roland Snooks observes that the algorithmic approach is an agent-based bottom-up approach where there is no predetermined idea of form, and form is dependent on the capability of the architect to 'encode architectural intent within the operation of the algorithm'.[23] As he explains, algorithms are used as generic templates for architects and they are 'abstract formal generators operating on an appropriated logic, devoid of any recognition of the architectural problem or proposition'.[24]

A change in the computational infrastructure of architecture

In 'Design Signals: The Role of Software Architecture and Paradigms in Design Thinking and Practice', Panagiotis Michalatos discusses how information technology, and therefore the complexity paradigm, altered architectural production through the inscription of the digital ontologies in architectural software.[25] He argues that the issue of what constitutes an architectural object is embedded in the data infrastructures of the software that architects use, and 'these ontologies determine what is observable, accessible, transmissible and achievable; in short, what is representable within a digital environment'.[26]

Practitioners within the fields of computer sciences and information technology design ontologies in order to deal with information and reduce complexity.[27] As Michalatos says:

> By designing an ontology, one determines the objects, operations and relationships that can be described within an information processing system. This determines what attributes are stored in files and databases, and what objects are presented to users to interact with. The makers of the software that architects use also therefore influence the design process and thinking, for they determine the objects and actions, the very language in which architects think while designing.[28]

After explaining the impact of computational infrastructures on design thinking and operations, he further argues that 'the more elaborate and specialised the ontology, the less suitable the software becomes for the early stages of design where ambiguity can be more productive'.[29] He then demonstrates this argument with the example of BIM software where there is a so-called *architectural ontology* with the presence of already defined architectural objects, such as walls, floors, staircases and doors etc. However, such an approach to design is already very limiting in the inscription of the design idea.[30] Especially for the early stages of design, where creativity is essential, the designer's intentionality radically drops with the enforcement of pre-defined objects and the increased elaboration of the software ontology. Consequently, design becomes limited first because the object of design and its associative tools are predetermined within the definition of such a specific ontology, and secondly, the data structures become partially accessible and interactive though the user interfaces that suggest 'a language through which specific aspects of a design can be considered'.[31]

The more complex design becomes, the more information is inscribed in an architectural object. Michalatos defines this condition as 'information granularity', resulting from the inscription and storage of massive data and tracking the network of actions that inform different parts and layers of the digital model, and therefore leading to highly

granular and distributed models 'to record and represent the design process itself and its outcomes'.[32] This inevitably encouraged architects to use workflows and their interfaces for the organisation of data and to make these highly granular models and data complexity accessible for themselves.[33]

In a recent issue of *Architectural Design*, Kutan Ayata, founding partner of the architectural office Young & Ayata, argues that the developments in computational technology has altered the evaluation and representation of architectural design process in two ways: 'form follows arrow' – step-by-step diagrams that try to make sense of formal transformations, and 'form follows data' – 'an overly redundant set of steps regarding the generation of form is displayed to demonstrate various software protocols, parameter performance, data inclusion and stages of digital maturation' with the aim of recording 'justifiable evidence of formal becoming'.[34] He criticises these practices for inevitably reducing the complexities of architecture into technological-looking linear representations and failing to reveal 'the logic of the system that [really] matters'.[35]

A change in the role of the architect

The discussion on the incompatibility of the nature of computation and associated methods with designers' intentionality and thought processes has another layer that further opens up a question about the role of the architect. While a number of scholars expect or call for a dissolution in architects' authoring design process, some still emphasise its necessity. This paper argues that this controversial status of the architect's role could be challenged by reconceptualising design intentionality. According to the writers of *Architecture and Authorship,* such an attempt will maintain 'a kind of topography for architectural action, therefore, forming a conceptual surface that allows architecture to develop as a coherent discipline.'[36]

Pablo Lorenzo-Eiroa raises a question on the role of the architect in authoring computational processes where the inner dynamics of software may inform the whole design formation: 'if structure is predetermined by the interface, the designer is merely interpreting a variation that completes the implicit combinations that the metaphysical project of the interface proposes, placing the programmer as the author'.[37] Here, structure or underlying logic corresponds to the inner principles of an interface that could easily affect the construction of formal logic and may restrain and determine the formal freedom and control of the architect. Hence, if the architects cannot express the individuality and encode design intentionality in algorithms, then the architects' role will be diminished, correspondingly, could be questioned and replaced by the programmer.

Scott Marble, editor of *Digital Workflows in Architecture,* points out another setback in the use of computational methods which is caused by the pre-determinacy of algorithms, and suggests an integration of computational control and architects' freedom to inscribe the design intentionality in algorithms, thus restating the architects' role and authorship.[38] He exemplifies David Benjamin's computational design approach as a resolution to this problem:

> Human intuition and judgment occur when designing the design space of a problem, by choosing the inputs and evaluating the outputs to an algorithm but also by designing the algorithm itself. This, then, is not seen as a reduction of authorship; by focusing exclusively on the design space as the locus for decision-making, algorithms are positioned as creative tools that expand the design capabilities of architects. By designing the algorithm, the relationship between constraints (to control possible design options) and variables (to explore possible design options) can become an integral part of the architect's overall design intent.[39]

As Marble informs us, Benjamin defines the role of the architect as the mediator between what

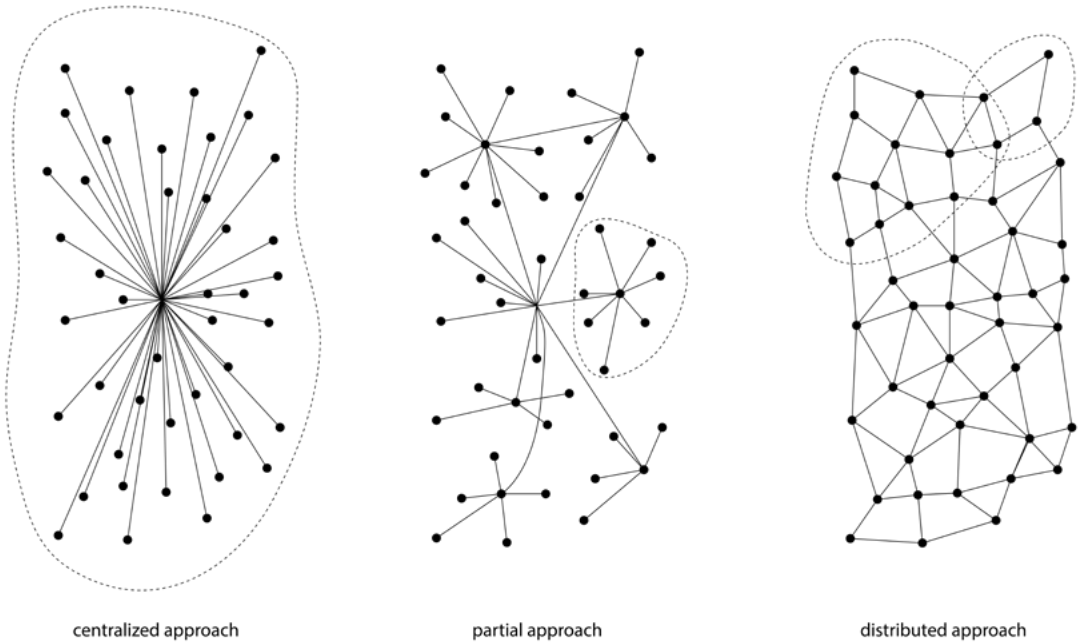

Fig. 1: Diagram of the tripartite model based on the network diagram proposed by Paul Baran, *On Distributed Communications*, (RAND Corporation, 1964), 2. Drawing: author.

is controllable and what is explorable in design processes; in this way the design intention could be inscribed and embedded into the algorithm, enabling freedom and control simultaneously.[40] Marble continues: 'the identity of the architect is largely built upon her or his ability to author design solutions' and the challenge is in 'capturing the full range of architectural design intent within digital workflows'.[41] He suggests the proper formation and expanded use of digital workflows, which has the potential to transform the role of the architect with freedom and control in computational processes that 'ha[ve] been increasingly displaced by technologically mediated processes over a long time'.[42]

David Benjamin criticises the position of the architect in using exiting software and programming languages: 'Yet algorithms are not neutral or inevitable. They are designed with assumptions and biases that condition what they produce. And if these assumptions were different, the designs produced through them would be different.'[43] In computational processes, if form generation is so dependent on the algorithms, and the designers cannot control them through their design intentions, the whole process and internal form relations would be delimited with the pre-determinacy of algorithms. Pablo Lorenza-Eiroa discusses this problem in his *Architecture in Formation*:

> It is quite clear that if architects do not recognize the underlying logic of the interfaces and displace the given source codes of algorithms to create their own, their work is trapped by a predetermined set of ideas, cultural projections, and aesthetic agendas contained within those interfaces.[44]

In order to trace possible negotiations between architects' intentionality and operational modes – or *inner structural dynamics* – of algorithms and computational processes, an organisational model is proposed.

An organisational model for architects' intentionality

In the field of computation and communication sciences, the concept of the network has been defined for the management and organisation of information.[45] In 1964, Paul Baran explored the possible hierarchical and non-hierarchical operational structures for communication and diagrammatised centralised, decentralised and distributed network models.[46] [Fig. 1] The concept made its way into the realm of architecture in the late twentieth century as 'network practice', which corresponds to the organisation and distribution of work and collaboration among design actors.[47] Tom Wiscombe writes about a complex organisational model – *emergent* networks – which is different from simple collaboration whose organisation is basically an accumulation around similar interests.[48] He suggests that emergent networks 'can create new and complex coherences out of divergent interests', whose products are non-predictable and non-linear.[49] It is possible to find a basis for these complex organisations in Baran's model in which different approaches to form computation can exist as independently, as well as combinatorial ones can be produced to map different forms of externalisations of design intentionality.

In the first diagram, the components are directly connected to the centre; accordingly, the only hierarchical layering is in between the centre point and the components. In the second model, the components are connected first to the local centres, and then these sub-centres carry the information to the main centre. In this decentralised approach, a multi-level hierarchical structure increases the chance of transmission of information compared to the first diagram. However, in the third diagram, there is no distinct organisational hierarchy and therefore each point could be assigned desired importance within the system. This distributed form of organisation simultaneously enables both freedom and control in the management of information.

Translation of this diagram and corresponding terms from communication and network sciences to the field of computational architecture as *centralised*, *partial* and *distributed,* constitutes a platform for an assessment of architects' intentionality in form computation.[50] Based on the employed computational logic and design intention, the possible interpretations of these approaches suggest a coherent *field* of recent approaches and methodologies to reconsider architects' intentionality in the computational form. Such mapping will provide a spectrum of approaches as well as exhibit a gradient epistemic scale, a representation in which the so-called epistemic oppositions – of subjectivity and rationalisation, human and computational thinking etc. – do not operate antagonistically (as competing notions), but rather, in a complementary manner.

Clearly, the approach to management and organisation of data is different in these three models. Through an analysis of the pattern formation and organisational structure between parts (inputs, outputs) and relations (design actions) within the whole design process, the overall computational approach can be assessed. [Fig. 2]

A centralised computational approach

The centralised approach can be described as the model where there is a single central node where all data is sent, which then directs the data to the intended recipient.[51] According to Alejandro Zaera-Polo, this kind of approach to computation can be interpreted as a centrally organised algorithmic system 'that tries to articulate everything at once'.[52]

In this approach, there exists an underlying idea about the formal logic rather than a predetermined idea of final form, and an algorithm can be designed or customised to write a specific code.[53] As Zaera-Polo notes, the condition that the use of computation is central to the generation of form makes this approach more vulnerable to the alterations in the selection and use of software, since any adjustment in the initiator or the structuring of the code will directly change the resultant configuration.[54] Therefore, in this model, there is a direct relationship between computational rationality and form generation which dominates and, in parallel, delimits the externalization of design intention. Therefore, this deeply nested relationship between the design of the computational structure and form generation entails a dependency on the ability of architects to translate design ideas into computable languages.

As an example, the work of Roland Snooks – whose approach falls into a rather experimental and innovative design field – reflects such a centralised model; in which, according to Mennan, he managed to inscribe his design intentions within the computational logic by participating actively in the computational processes through what Snooks calls 'strange feedback' that attempts to hybridise creative characteristics of both bottom-up algorithmic processes with the top-down decision mechanisms of architects.[55] Such interference to centralised algorithmic systems is very difficult to employ; since it requires relying, on the one hand, on to a high level of expertise in computational design methods, and on the other, on an ability to understand the nature of their limitations and find ways to incorporate them. Despite this challenge, similar attempts will extend the creative capacity of the computational design processes, this time enabling more control and freedom to the architect as well.

A partial computational approach

The second model in Paul Baran's diagram corresponds to a decentralised networking system, in which there exists a hierarchy between parts, subcentres and the centre: a series of subcentres are connected to the main centre; rather than the heaviness of one-centred system.[56] This condition of having multiple centres/processors instead of a single one enables a specialisation between

the subcentres and its connected part. In such a condition of compartmentalisation, each cluster is expected to work in itself, and later, the outcome is transferred from a subcentre to the main centre. Then, all data is processed in the main centre. In this approach, manual and computational design processes can be combined. The central organisational system could be a computational structure or a conventional design process. Since in this model, design could be composed of multiple methods that are partially processed either intuitively or determined by computational rationality in sub-centres, the term 'decentralised' is assessed and interpreted as 'partial' in the context of computational architecture. Such a terminological shift is necessary to reflect certain approaches and intentions to use computational methods and associative technologies, and then, open up further discussions on designers' intentionality in computational architecture.

Partial computation is already a functional method in the field of computer science, used in the evaluation and optimisation of partial programmes with the given parameter values.[57] If we borrow and apply this definition to the field of architecture, it suggests the application of computational methods to evaluate and optimise *partial phases* of the design process with the given parameter values, where computation is not necessarily central, but rather partial to formal content and overall organisation.

According to David Benjamin, studies based on optimisation and efficiency can be placed under this approach, where the main reasoning in the use of computation is not *exploratory,* but rather *explanatory*.[58] Based on this definition, pre-rationalisation and post-rationalisation can be discussed as the dominating uses of this approach and therefore positioned under a partial computational model with reference to their partial capacity to have an impact on the overall design approach and form generation.

It can be argued that the intentionality in the use of computation is similar in both approaches, however in pre-rationalisation, computational rationality is superior to architects' subjective thought processes, whereas in the post-rationalisation, subjectivity and intuitive decision mechanisms are prioritised. In the former, the design process begins with a deterministic approach, whereas for the second approach, the formal logic has the flexibility to be dominantly subjective and intuitive, yet eventually, is partially rationalised to evaluate and optimise the intended form for fabrication or for performative reasons.

As a critique of this performative or optimisation approach to form computation, Benjamin examines efficiency and creativity, two contrasting yet complementary concepts, and their implications in the field of architectural design. He names them, 'exploitation' and 'exploration', meaning respectively, 'utilising [the] existing' and 'searching for [the] new'.[59] He states that

> designers interested in exploitation prefer a narrow, continuous design space, such as a slanted plane or a topological surface with one or two bumps. In this case, it is possible to quickly hone in on the region of best performance and to locate the single global maximum. The simpler the design space is, the faster they can find the optimal design.
>
> Designers interested in exploration prefer a wide, discontinuous design space, such as a jagged mountain range with multiple peaks. In this case, there are many distinct regions of good performance, and it is often possible to find multiple local maximums that are both interesting and high-performing, even if they are not the global maximum. The more complex the design space is, the more likely it is that they will make an unpredictable discovery.[60]

Benjamin also suggests introducing 'subjective criteria' into optimisation processes in order to integrate the seemingly separate qualities of

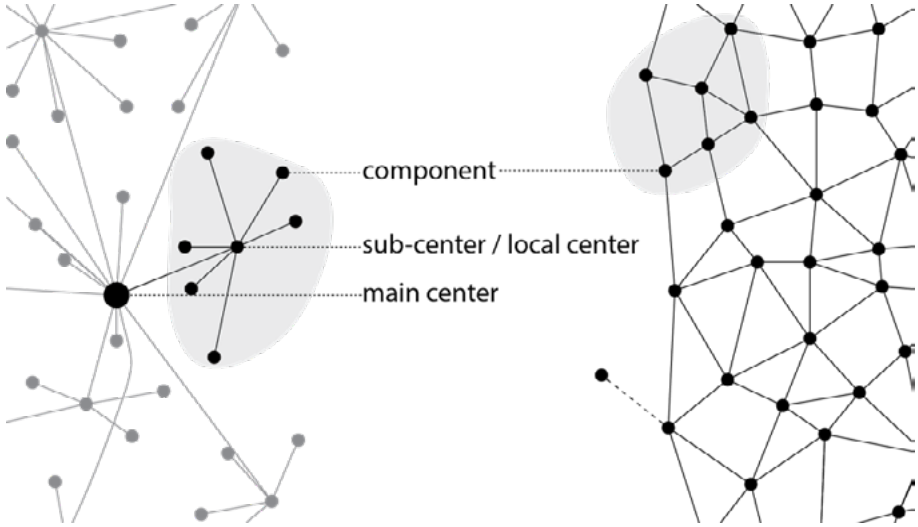

Fig. 2: Varying part-whole relationships in computational models. Image: author.

human intuition and creativity with computational thinking.[61] Even though such a method is under-utilised, it would enable designers to incorporate subjective criteria, such as aesthetics, mood, identity and interpretation of architectural programme, with objective technical criteria, like structural performance and circulation efficiency, in the same optimisation process. In such a process, he argues that the subjectivity of the architect is translated into objectives and value judgment, and the designer's creativity comes from 'designing objectives and designing experiments rather than simply designing solutions', makes the architect more engaged in designing the problem and focused on potential design space, 'the complex topological surface'.[62] About the degree of subjectivity in these processes, he claims, 'although they might be buried and hidden, they are there.'[63]

It is possible to place pre-rationalisation, post-rationalisation and reverse engineering under this model. As the name clearly expresses, in *pre-rationalisation*, the rationalisation process is at the early stages of form generation, consequently the formal logic is dependent on the initial-factual data and therefore, arguably, highly objective. This approach can also be defined as a data-centric approach since the form is optimised from the beginning, and efficiency is the major decisive factor in form generation.[64] Thomas Fischer explains this approach by mentioning Buckminster Fuller: '[his] approach of addressing design challenges before they become acute, which he referred to as "comprehensive anticipatory design science" is largely based on the concept of pre-rationalisation'.[65]

As a result of the dependency of form on the data, the freedom and subjectivity of the architect in form generation can be evaluated as *low*, but since the construction of the design problem and the intention to use such methods belong to the architect, it still embodies some degree of subjectivity but in a highly rationalised form.

In *post-rationalisation*, on the other hand, the use of computation is partial; the rationalisation process is placed at the final stages of form generation. The formal logic is dependent on the intuitive and artistic decision making of the architect and therefore, arguably, belongs to the 'subjective world of states of consciousness, or of mental states – with intentions, feelings, thoughts, dreams, memories'.[66] This approach can also be referred to as the intuitive approach, since intuition is the major decisive factor and source of reason in form generation. Although intuitive processes depend more on subjective design decisions, this approach still requires a degree of rationalisation at the final stages in order to calculate structure and to construct and fabricate the final form.

Based on the intuitive decision-making of the architect, this approach mostly denotes a traditional top-down approach where the creator relies on his/her background knowledge and former experiences. This 'knowledge-based approach, as William Mitchell defines it, can be problematised, as the design intentions and mechanisms are inaccessible since they exists in a closed system or a 'black box' where the idea of form is in the designer's mind and is predetermined.[67]

There exists a more generative version of post-rationalisation, which deals with *reverse engineering* in order to unfold the black box and translate the subjectively constructed form to a computable environment through extracting the underlying logic and geometry of a final form. By doing so, this method enables more than just post-rationalisation; it helps to breed new variations of the reverse-engineered form from unfolding its design mechanisms.[68] The biggest challenge here is the involvement of a secondary subject who is not the author of the design, but another designer who interprets the process – and his/her ability to understand the design intentions and formal logic and translate them into a computational model.

Mark Burry's research on Gaudí's design of the Sagrada Familia is an instructive example of this type of approach.[69] Here, according to Neil Leach, Burry explores 'digital techniques for understanding the logic of Gaudí's own highly sophisticated understanding of natural forces'.[70]

One may remark that the partial approach to computation is highly practical and offers more freedom to designers in the employment of the design intentionality. However, it fails in rendering a generative and creative formal approach, as it lacks exploring the potentials brought along by the computational world. The idea here is much rather to confirm the design decisions with calculations and validating the final form, instead of creating a new ground for unprecedented forms and formal relations.

A distributed computational approach

In the simplest version, distributed computation can be defined as the condition where the multiple use of algorithms and codes is distributed through the different and particular stages of the design process. Different from partial computation, this approach includes both design exploration and exploitation, and furthermore, it is flexible and intention-oriented.[71] Therefore, in this approach, intentionality is distributed among the multiple human and non-human agencies, and as Alejandro Zaera-Polo explains, it is 'the co-evolution and optimisation of relationships between multiple routines, mediated through the mainframe, which is able to produce real innovation, rather than the heaviness of a centrally organised system that tries to articulate everything at once'.[72]

This approach can also be referred to as a non-linear workflow approach in which computational methods are used and customised to a certain degree, to adopt the architect's intentionality. It includes employing custom and disposable codes, which are 'intentionally purpose-built for the task at hand', to respond to specific problems or spontaneous needs at certain phases of the design process in order to encourage creative thinking rather than perfect the code itself.[73]

Promoting a distributed approach in computational design, Tom Wiscombe explains the problem with a centralised computational approach: 'You lose too much information when everything in an architectural problem has to be processed through an algorithm. Inputs are forced to become quantitative or otherwise abstract in order to be able to be computed, so it is not surprising that outputs are also anemic'.[74] Wiscombe further criticises the categorisations of design approaches based on dichotomies such as bottom-up or top-down and suggests the implementation of the right position and a useful design tool for the problem.[75] He states that

> there are such hardened camps now: you are either a bottom-up researcher or a top-down designer; you either experiment with means, or you design towards ends. A crossover term I like is 'messy computation' – it is open-ended enough to allow you to be a designer but also capitalises on the advantages of recursion and agency. Nothing is taboo that way. You pick and choose the right tool for the job, and more importantly, create custom workflows which jump around between techniques. It's a patchwork of scripting, modeling, painting, and engineering, which I find very convenient, and happily, free of ideology.[76]

In 2009, Neil Leach points out a shift prior to the introduction and multiplication of computational methods in architectural design, explaining that 'the architectural imagination has been displaced into a different arena – into the imaginative use of various processes.'[77] Calling for a change in the dominant approach to computation, in 2012, Scott Marble identifies a further shift experienced in the dominant computational approach: 'from process to workflow'.[78] He states that 'the identity of the architect is

largely built upon her or his ability to author design solutions' and the challenge is in the 'capturing the full range of architectural design intent within digital workflows', and he suggests the proper formation and expanded use of these non-linear workflows, that have the potential to restate the architects' intention with the freedom and control that have been 'increasingly displaced by technologically mediated processes over a long time'.[79]

In a recent issue of *Architectural Design*, Richard Gerber indicates the inclusiveness of workflows as 'they can accommodate the personalized design processes of architects as well as integrated engineering strategies and collaborative ideas about building delivery'.[80] Along with the inclusive nature and flexibility of a non-linear workflow approach, the challenge is the management of the complexity created by the involvement of different specialised design parts and the vast amount of specialised information distributed within the same process.[81] However, it allows for design collaboration and inclusive design practices in order to respond the increased amount of information and complex design problems. This leads to a great extension in the capacity of the architect. Accordingly, this dismantling and distributing of computation throughout the design process makes it more efficient and flexible to encode the designer's intentionality. Hence, the design intentionality becomes more visible due to the radical increase in freedom and control over form. Furthermore, it enables instantaneous design experimentation and rationalisation within the same system by integrating two seemingly contrasting worlds – that of computation and intuition – with more freedom in subjective criteria.

Conclusion

Presenting a theoretical framework to reconceptualise architects' intentionality in computational form generation processes, this paper instrumentalises communication and network diagrams as an alternative reading of current approaches to form computation. The point of departure in this search arises from the gap emerging from the epistemological opposition of human and computational thought processes, and in turn, the changing role of the architect in computational design. This gap created by the shift to the language of computation and its associated rationality requires reestablishing the modes of intentionality, since as Mennan suggests 'calculation leaves an incomplete space that cannot be saturated with information alone and waits to be filled with meaning and interpretation'.[82] Recent attempts reveal that such reconciliations are possible. A new model in which design intention is encoded within the operation of code writing is in the process of replacing the dominant computational model. Such a change is indicated in these new approaches where intentionality remains impure, distributed and embedded within the computational models.

Notes

This article has stemmed from my PhD study at METU, Department of Architecture supervised by Prof. Dr. Zeynep Mennan, to whom I would like to express my sincere gratitude. A portion was developed during my research at Columbia University GSAPP as a Fulbright visiting scholar, and an earlier version of the study was presented at the CAAD Futures Conference and published in the book of proceedings in June 2017.

1. Zeynep Mennan, 'Mind the Gap: Reconciling Formalism and Intuitionism in Computational Design Research', *Footprint* 15 (2014): 33.
2. Ibid.
3. Humberto Maturana and Daniel Rosenberg, 'Design as Doing: A Conversation with Humberto Maturana about What Designers Do', *Dosya 29: Computational Design* (2012): 20.
4. Ibid., 23.
5. Ibid., 20.
6. Ibid., 23.
7. Ibid.

8. Ibid.
9. George Stiny, 'What designers do that computers should', in *The Electronic Design Studio: Architectural Education in the Computer Era*, ed. Malcolm McCullough et al. (Cambridge, MA: MIT Press, 1990), 17–30.
10. Ibid.
11. Ibid., 19.
12. William Mitchell, foreword to Kostas Terzidis, *Expressive Form: A Conceptual Approach to Computational Design* (London: Spon Press, 2003), vii.
13. Ibid.
14. See Terzidis, *Expressive Form*.
15. Ibid., 6.
16. Ibid.
17. Ibid., 29.
18. bid., 6.
19. Ibid., 4.
20. Axel Kilian, 'Computational Design as a Process to Support Design Exploration rather than Design Confirmation', *Dosya 29: Computational Design* (2012), 44.
21. Ibid.
22. Ibid., 45.
23. Roland Snooks, 'Volatile Formation', *Log* 25 (Summer 2012).
24. Ibid.
25. Panagiotis Michalatos, 'Design Signals: The Role of Software Architecture and Paradigms in Design Thinking and Practice', *Architectural Design,* Vol. 86 (2016): 109–115.
26. Ibid., 110.
27. Ibid.
28. Ibid.
29. Ibid.
30. Ibid.
31. Ibid., 111.
32. Ibid., 110.
33. Ibid., 115.
34. Kutan Ayata, 'Ruptured Flows: An Argument for Nonlinear Workflows', *Architectural Design*, vol. 87 (2017): 92.
35. Ibid.
36. Tim Anstey et al., Introduction in *Architecture and Authorship,* ed. Tim Anstey, Katja Grillner and Rolf Hughes (London: Black Dog Publishing, 2007), 6.
37. Pablo Lorenzo-Eiroa, 'Form in Form: On the Relationship between Digital Signifiers and Formal Autonomy', *Architecture in Formation: On the Nature of Information in Digital Architecture,* ed. Pablo Lorenzo-Eiroa and Aaron Sprecher (London: Routledge, 2013), 19.
38. Scott Marble, 'Editor's Notes: Authorship', in Marble (ed.), *Digital Workflows in Architecture: Design–Assembly–Industry* (Basel: Birkhäuser, 2012), 26.
39. Ibid.
40. Ibid., 26–27.
41. Ibid., 27.
42. Ibid.
43. David Benjamin, 'Beyond Efficiency', in Marble, 'Editor's Notes', 16.
44. Lorenzo-Eiroa, 'Form in Form', 19.
45. See Albert-Laszlo Barabasi, *Linked: The New Science of Networks* (Cambridge: Perseus Publishing, 2002).
46. Paul Baran, *On Distributed Communications*, (The RAND Corporation, 1964), 47. Tom Wiscombe, 'Emergent Models of Architectural Practice', *Perspecta* 38 (2006): 60.
48. Ibid.
49. Ibid.
50. As an indirect translation from Baran's diagram, I propose the term 'partial' instead of 'decentralised' for a better definition.
51. Baran, *On Distributed Communications*, 1.
52. Alejandro Zaera-Polo*, The Sniper's Log: Architectural Chronicles of Generation X* (Barcelona: Actar, 2012), 443.
53. Terzidis, *Expressive Form,* 69.
54. Zaera-Polo*, Sniper's Log*, 443.
55. Mennan, 'Mind the Gap', 41.
56. Baran, *On Distributed Communications*, 2.
57. Yoshihiko Futamura et al., 'Essence of generalized partial computation', *Theoretical Computer Science,* vol. 90, no. 1 (1991): 61–79.
58. Benjamin, 'Beyond Efficiency', 14–25.

59. Ibid., 15.
60. Ibid., 15–16.
61. bid., 22.
62. Ibid.
63. Ibid., 23.
64. Thomas Fischer, 'Geometry Rationalisation for Non-Standard Architecture', *Architecture Science* no. 5 (2012): 40.
65. Ibid.
66. In discussing 'knowledge' and 'imagination' as objective and subjective oppositions, Kostas Terzidis refers to Karl R. Popper (The Logic of Scientific Discovery, 1968) in Terzidis, *Expressive Form*, 73.
67. William Mitchell et al., 'Top-Down Knowledge-Based Design', in *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era,* ed. Malcolm McCullough et al. (Cambridge, MA: MIT Press, 1990), 137–148.
68. Eldad Eilam, *Reversing: Secrets of Reverse Engineering* (London: Wiley, 2005).
69. Mark Burry, *Scripting Cultures: Architectural Design and Programming* (London: Wiley, 2011).
70. Neil Leach, 'Digital Morphogenesis', *Architectural Design,* vol.79 (2009): 35.
71. Zaera-Polo*, Sniper's Log*, 443.
72. Ibid.
73. Marty Doscher, 'Disposable Code: Persistent Design', in Marble, *Digital Workflows*, 207.
74. Interview with Tom Wiscombe by Ralf Broekman and Olaf Winkler, *Build Das Architekten Magazin* (March 2010).
75. Ibid.
76. Ibid.
77. Leach, 'Digital Morphogenesis', 35.
78. Scott Marble, 'Introduction, From Process to Workflow: Designing Design, Designing Assembly, Designing Industry', in Marble, *Digital Workflows*, 7–11.
79. Ibid.
80. Richard Garber, 'Digital Workflows and the Expanded Territory of the Architect', *Architectural Design*, vol. 87 (2017): 13.
81. Marble, 'Introduction', 7–11.
82. Mennan, 'Mind the Gap', 40.

Biography

Having received her undergraduate and graduate degrees from the same department, Duygu Tüntaş is currently a PhD candidate and full-time faculty member at Middle East Technical University (METU) Department of Architecture, in Turkey. Her research interests include architectural form processes and computational design research. She has been awarded several prizes in architectural design competitions and received a Fulbright visiting scholarship for her PhD research.